Experience in using commercial clouds in CMS

L Bauerdick[1], B Bockelman[2], D Dykstra[1], S Fuess[1], G Garzoglio[1] , M Girone[3] , O Gutsche[1], B Holzman[1], D Hufnagel[1], H Kim[1], R Kennedy[1], D Mason[1], P Spentzouris[1], S Timm[1], A Tiradani[1] and E. Vaandering[1]

[1] Fermi National Accelerator Laboratory, PO Box 500, Batavia IL 60510-5011
[2] University of Nebraska-Lincoln, Lincoln, NE 68588
[3] CERN, 1211 Geneva 23, Switzerland

on behalf of the CMS collaboration

Abstract. Historically high energy physics computing has been performed on large purpose-built computing systems. In the beginning there were single site computing facilities, which evolved into the Worldwide LHC Computing Grid (WLCG) used today. The vast majority of the WLCG resources are used for LHC computing and the resources are scheduled to be continuously used throughout the year. In the last several years there has been an explosion in capacity and capability of commercial and academic computing clouds. Cloud resources are highly virtualized and intended to be able to be flexibly deployed for a variety of computing tasks. There is a growing interest amongst the cloud providers to demonstrate the capability to perform large scale scientific computing. In this presentation we will discuss results from the CMS experiment using the Fermilab HEPCloud Facility, which utilized both local Fermilab resources and Amazon Web Services (AWS). The goal was to work with AWS through a matching grant to demonstrate a sustained scale approximately equal to half of the worldwide processing resources available to CMS. We will discuss the planning and technical challenges involved in organizing the most IO intensive CMS workflows on a large-scale set of virtualized resource provisioned by the Fermilab HEPCloud. We will describe the data handling and data management challenges. Also, we will discuss the economic issues and cost and operational efficiency comparison to our dedicated resources. At the end we will consider the changes in the working model of HEP computing in a domain with the availability of large scale resources scheduled at peak times.

1. Introduction

Every stage of a modern HEP experiment requires massive computing resources (compute, storage, networking). Detector and simulation-generated data have to be processed and associated with auxiliary detector and beam information to generate physics objects, which are then stored and made available to the experimenters for analysis. In the current computing paradigm, the facilities that provide the necessary resources utilize distributed High Throughput Computing (HTC), with global workflow, scheduling, and data management, enabled by high-performance networks.
The evolution of the HEP experimental program (upgrades, new experiments) will generate increased computing needs on the timescale of 2025 that are expected to go well beyond any capacity increases expected from Moore's law or advancements in computing architectures.

The increased precision, event complexity, and luminosity of the high luminosity phase of the Large Hadron Collider (HL-LHC) alone will push computing needs nearly two orders of magnitude above current HEP capabilities, while generating exabytes of data.

It is essential for HEP to develop the concepts and deploy the infrastructure that will enable analysis of these vast amounts of data efficiently and cost-effectively. The industry trend is to use "Infrastructure as a Service" through Cloud computing, to reduce cost of provisioning and system operations, to provide redundancy and fail-over, to rapidly expand and contract resources (elasticity), and to pay only for the resources needed/used.

The HEPCloud Facility [1] concept is envisioned to be a portal to an ecosystem of computing resources---commercial or academic---that will help our facilities move away from standalone solutions. It will provide ``complete solutions" to all users, with agreed-upon levels of service, routing user workflows to local (``owned") or remote (``rental") resources based on efficiency, cost, workflow requirements and the policies of the facilities hosting the resources. The goal is to integrate rented resources into the current Fermilab computing facility in a manner transparent to the user. The first type of external resources considered was commercial clouds, partnering with Amazon Web Services as the provider. For our studies, we identified use cases that both demonstrate the necessary aspects of the concept, and that are also useful to the experimenters. One of the use cases focused on CMS Monte Carlo generation and reconstruction, targeting physics results for the Moriond conference in March 2016. This use case studied the scalability and sustainability of elastic provisioning of AWS resources through the portal, and exercised the prototype decision engine and cost model.

2. Goals of the CMS AWS project

The CMS experiment is confronting a large and ever-increasing computing challenge, even before the start of the HL-LHC. To meet the growing computing needs, CMS has investigated use of resources beyond the traditional grid-provided systems. One potential area of growth is in dynamically provisioned resources---either via academic and opportunistic access, or through commercially provided computing services. Beginning in 2015, CMS began to seriously explore commercial cloud provisioned resources. The logical platform choice was the market leader [2] in Cloud Infrastructure-as-a-Service, Amazon Web Services (AWS).

In the past, CMS has demonstrated small-scale cloud computing for a short amount of time as a proof-of-concept to investigate feasibility. This demonstration does the next big step and is intended to show the ability to increase the global processing capacity of CMS by a significant fraction for an extended period. Importantly the test was also intended to demonstrate the capability of executing in production any of the CMS centrally organized workflows on AWS. Additionally, the project intended to deliver useful simulated physics events to the collaboration for analysis at a production scale for the physics conference Moriond 2016. CMS was awarded a 9 to 1 matching grant from AWS that allowed the purchase of $300k of credits for computing, storage, and network charges for an investment of $30k. The size of the award was based on an estimate of what it would cost to do one month of large-scale processing.

Additionally, a conditional cost waiver was granted for exporting data; as long as the export costs remained under 15% of the total monthly bill, and the produced data were transmitted across research networks such as ESNet, the export charges would be waived entirely. This discount program was so successful that it has been extended to researchers at all academic and research institutions [3].

In this paper, we will discuss the tests performed, the services required, the scale and performance achieved. We will demonstrate the ability to significantly augment the peak capacity of the CMS production system using commercial computing resources, investigate scheduling for peak, cost, reliability and efficiency of large-scale commercial computing resources and investigate the capability of the CMS experiment workflow management system to scheduling for peak.

3. Scheduling for peak

The computing resources for HEP experiments are pledged yearly by the funding agencies of countries participating in WLCG. Because of the lead time to commission physical resources in the computing centres in the different countries, the planning process looks 18 months ahead. The HEP experiments plan and request computing resources yearly. Requests are scrutinized by the Computing Resource Scrutiny Group (C-RSG) and eventually endorsed by the LHC Experiments Resources Review Board (RRB) in a formal process. All pledged resources are then made available throughout the year and it is very important that the experiments' central production teams plan for steady and continuous use during long periods of time, as shown on the left side of Figure 1.

Experience from Run1 and Run2 at the LHC shows that the computing needs of experiments are not constant over time. A number of activities, such as data (re)processing, simulation data generation and reconstruction, tend to come in bursts with irregular time structure, dictated by software release, conference and data taking schedules. In the example of a conference deadline, production activities have to start well in advance to make the deadline with constant resource usage. Instead, elastic resources could enable a much more compressed processing plan starting shortly before the conference. The available elasticity of bursting resources into commercial clouds would change the way people work in large scientific collaborations and allow for shorter and more agile time schedules [4]. The right side of Figure 1 shows this case where processing and simulation is done in burst. With resources provisioned with commercial clouds, the planning process could also be condensed. Time to provision resources is shorter because physical resources don't have to be provisioned and installed at the computing centres.

Provisioned resources like AWS may also provide a powerful source for problem recovery. In case of a problem that invalidates work already performed (a software problem, a systematic computing problem, or a problem of conditions) there is not sufficient excess capacity in the system to perform the work twice, without having to make difficult choices to cancel needed future work. At the same time it is not possible in the current budget environment to reserve excess capacity to recover from problems. The cloud model is interesting because it allows for the dynamic purchase of sufficient capacity to solve problems without maintaining dedicated resources in reserve. This ability to burst to a high fraction of the total CMS resources for a period of time should be seen as a useful insurance policy to recover from problems.

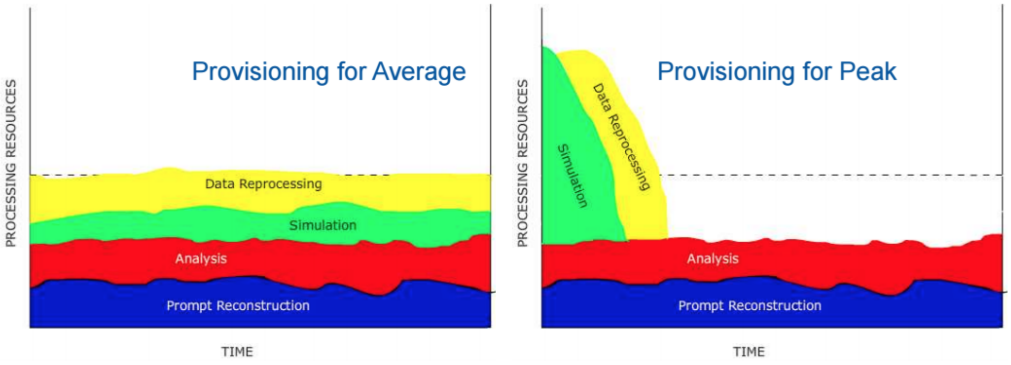

Figure 1. An illustration of provisioning for average vs. provisioning for peak.

4. Workload and workflows

CMS evaluated several production workflows based on a variety of criteria: physics value and current needs of the experiment, difficulty of the workflow, the potential for gaining knowledge by operating cloud provisioned resources, and the most effective use of the resources (given the potential for additional charges for staging out data).

The four workflows considered were: GEN-SIM, Data-RECO, DIGI-RECO, and GEN-SIM-DIGI-RECO. Generation and Simulation (GEN-SIM) is the simplest because it requires only parameters as input, and it runs CPU-bound on a single core. As the simplest workflow, there is not much to learn from it and CMS did not have a need for a large number of new GEN-SIM events, so this workflow was not favoured. Data reconstruction (Data-RECO) relies on raw events served by a global data federation and produces output where every job must be completed to maintain the integrity of the dataset. CMS did not have an urgent need for Data-RECO, and there was some concern about using an untested infrastructure for a mission-critical task, so also this workflow was not favoured.

Event crossing simulation and reconstruction (DIGI-RECO) adds several challenging elements: the input data from the previous step must be served over the global data federation, but the additional minimum bias events to create the full crossing need to be served from local storage in the CMS ``classic" mixing scenario. Each event reads hundreds of minimum bias ``pile-up data" events. Reading these events over the wide area network would result in very low CPU efficiency. In the end, a sizable sample of minimum bias events was placed on Amazon S3 storage to be accessed directly from the production jobs. The DIGI-RECO workflow was a consideration because of the technical capability that could be demonstrated. In the end, the workflow that was used for the majority of the processing was the workflow that combined all the elements into a single chain: GEN-SIM-DIGI-RECO. It is slightly simpler than DIGI-RECO, because it requires only parameters as input, but it has a higher amount of processing per event as it includes the generation and simulation steps. The higher amount of needed CPU time results in a more favourable ratio of processing to output event size, which allowed CMS to remain within the 15% data egress waiver. GEN-SIM-DIGI-RECO is a single complete workflow and several high impact samples were identified to demonstrate the value of the AWS resources. The individual pieces of GEN-SIM-DIGI-RECO refer to the four steps in jobs themselves, each an invocation of the CMS Software framework (CMSSW) processing the results of the step prior to it:

i. GEN: Madgraph 4-vector event generation

ii. SIM: Propagation of particles through the GEANT detector simulation

iii. DIGI: Electronics simulation and event digitization, including mixing in pile-up data pre-staged to S3 storage

iv. RECO: Reconstruction of the data into physics quantities used in analysis

Only the products from the last step were kept and staged out to the Fermilab EOS storage system, namely the AOD and the MINIAOD. In past production campaigns, CMS typically executed GEN, SIM, and DIGI-RECO as separate sets of jobs. Each job set involved stage-out of the intermediate data, sometimes restaging from tape, namely I/O activities which would increase the production costs at a resource like AWS. Since all the CMS software framework releases for the different steps of the workflow in question were in production and available, work was put into deploying these steps all into a linked chain of jobs on the workers themselves, called a step-chain. For the large-scale run, four large Standard Model background samples were produced in full chain GEN-SIM-DIGI-RECO workflows. The AWS S3 storage system would have been able to handle the load of the classic mixing scenario of CMS reading hundreds of minimum bias event objects to simulate a crossing, but cost optimizations required the minimum bias files to be staged to local storage attached to the VMs.

5. Scale, costs and stability of the services

AWS sells their excess resource capacity following a market model called ``Spot Market''. For every combination of machine type, availability zone, and region, users supply a bid price that represents the maximum that they are willing to pay per hour of computing time. AWS sets a dynamically changing ``spot price'' based on the current supply and demand. If the user's bid price is above the spot price, and there is sufficient capacity in the resource pool, the resources are provisioned at the spot price. If the spot price fluctuates above the bid price after a resource has been provisioned, the user is pre-empted with a two-minute advance notice. Resources are charged on the hour boundary and when the instance is terminated by the user; in the event of pre-emption, the last fraction of an hour is not charged.

The glideinWMS workload management system [5] was used to provision the worker nodes used during the CMS run. A development version of glideinWMS was deployed to make available some of the new features needed to run at scale.

Motivated by a previous study [6], a simple bid strategy for spot pricing was selected, which was to bid 25% of the ``on-demand'' price for a given resource. Figure 2 shows the ramp up in terms of number of job slots, from an initial 10%, to 25% up to the full 58000 slots (single core equivalent).

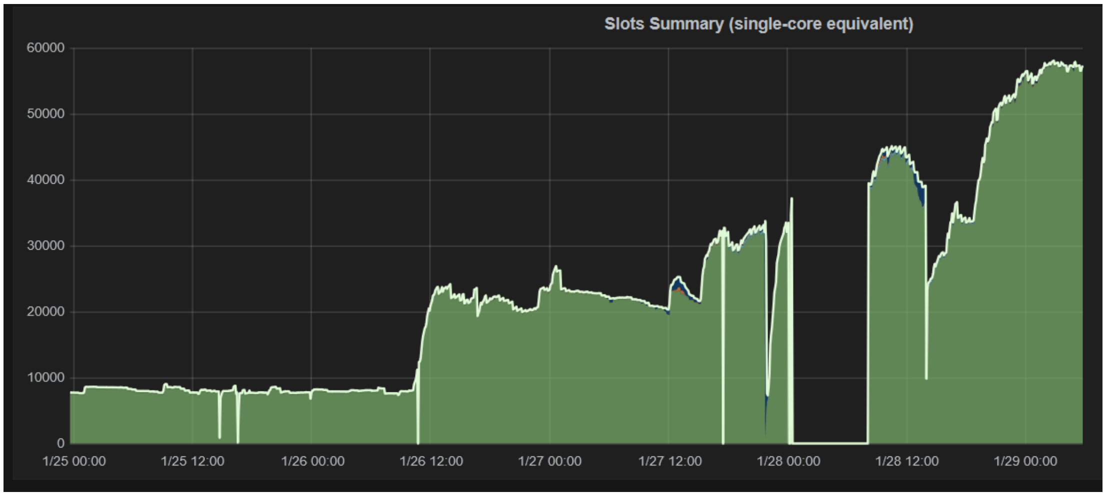

Figure 2. Ramp up of the AWS scale test from 10% of the total resources to full scale.

In this study, the static bid of 25% was found to perform as well or better than various adaptive algorithms previously described in the literature. In order to ``diversify our portfolio'' and improve the availability and stability of the system at scale, we bid in more than 100 different spot markets, representing nearly all the regions and zones then available in the US, as illustrated in Figure 3. The mean lifetime for a provisioned resource was 37.6 hours allowing it to run multiple individual jobs, while the average job lifetime was 4.7 hours. Figure 4 shows the distribution of provisioned resource lifetimes. While the distribution is peaked in the lowest bin, the tail is very long--some resources remained in the pool for over 200 hours.

About 2.9 million jobs were completed, equivalent to 15.1 million wall hours. 9.5% job failures, including pre-emption from spot market. A total of 518 millions events were generated, with 87% CPU efficiency. The tests performed by Fermilab and CMS on AWS have demonstrated that it is possible to utilize dynamically provisioned cloud resources to execute many CMS workflows at large scale. As shown in Figure 5, **the HEPCloud Facility was able to increase the amount of resources available to CMS by 33%**. .

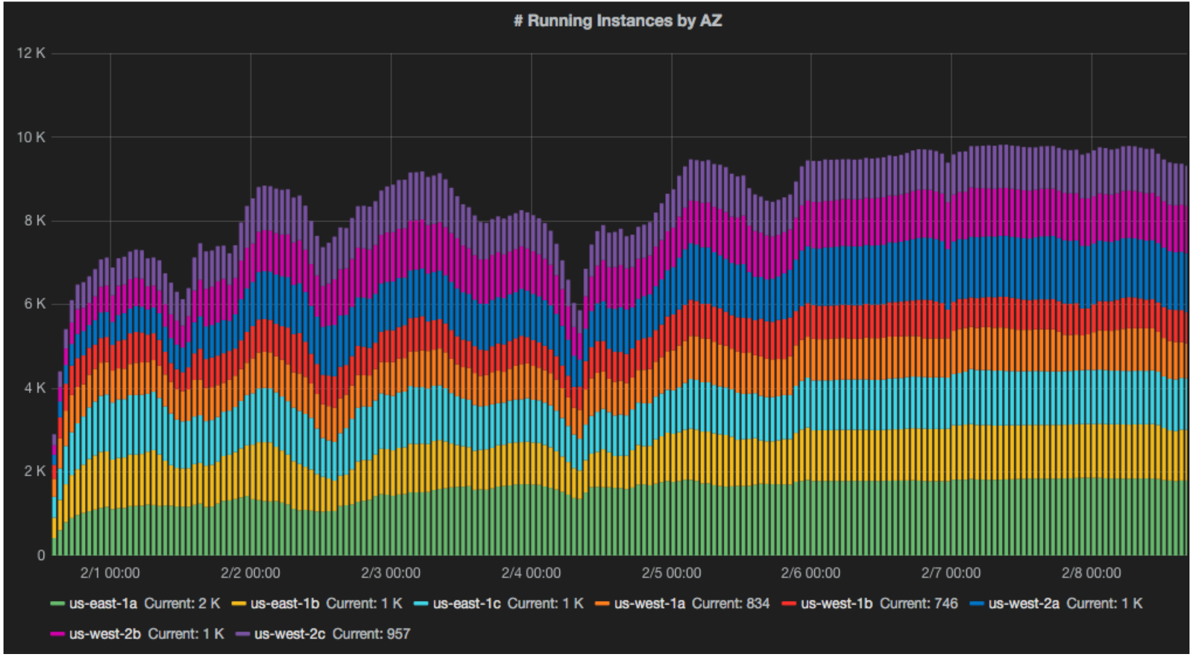

Figure 3. Number of running instances by AWS region and availability zone.

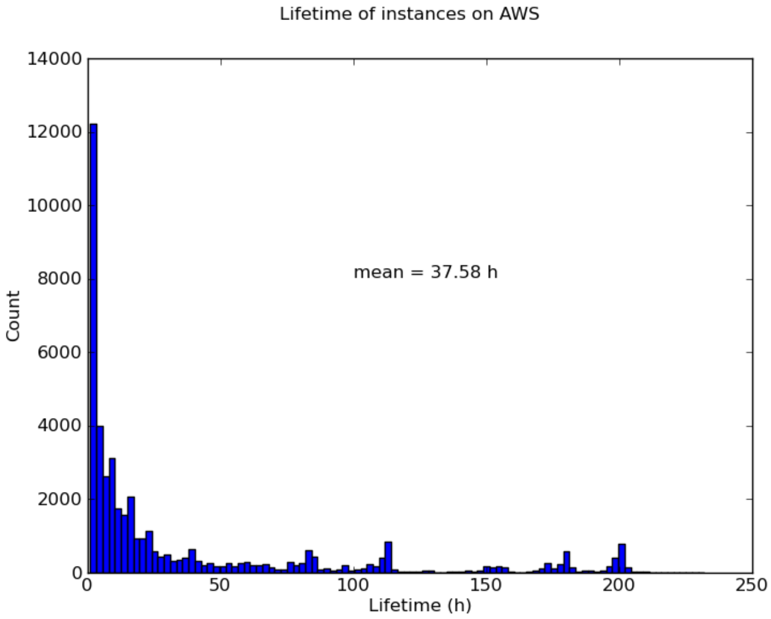

Figure 4. Pilot lifetime in hours.

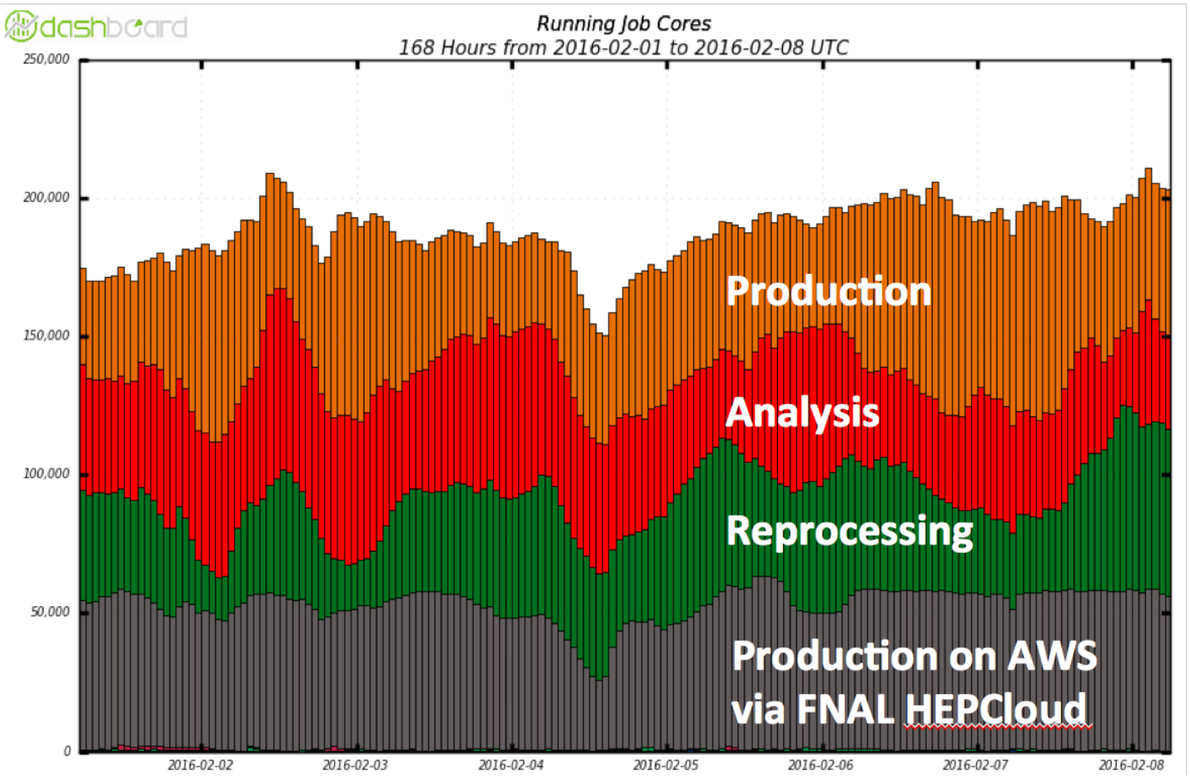

Figure 5. A comparison of the scale of processing on AWS to other global CMS activity.

6. Cost comparisons

Fermilab attempted to estimate the cost per core hour of the CMS Tier-1 processing resources as reported in detail in [1]. In order to compare core cost on must ensure the core perform similarly well. To make this comparison the CMS experiment simulates ttbar events on all hardware being compared and ensures roughly the same number of events are produced in the same interval of time. The estimate assumes 100% utilization of Tier-1 resources; at lower utilization, the effective cost per productive CPU cycle is larger. Given the uncertainty in the subjective inputs, the estimated error on the per hour core cost is roughly 25%. Results are reported in Table 1.

Table 1. The cost per hour for one core of computing on dedicated Tier-1 resources at Fermilab and on virtualized commercial cloud resources on AWS and the *ttbar* benchmark (greater = faster). The uncertainty in the AWS cost data corresponds to one standard deviation from the daily cost per core-hour.

	Average cost per core hour	*ttbar* benchmark
Fermilab CMS Tier1	\$0.009 ± 25%	0.163 *(ttbar/s)*
AWS	\$0.014 ± 12%	0.158 *(ttbar/s)*

7. Conclusions

The HEP experimental program continues to evolve, and will require computing capacity in excess of current and future on-premises resources. A sensible target which will yield the most benefits is to leverage the industry trends in cloud computing. The HEPCloud Facility concept provided CMS a portal to these computing resources as a transparent layer for the experiment. The goal of the CMS experiment was to enable the execution of a physics workflow that would add significantly to not only their overall global resource consumption but also generate useful analysis results for the collaboration. For a full simulation workflow (from event generation to physics reconstruction), over 15 million hours of computing was consumed, simulating more than 500 million events. The steady-state cost came to 1.4±12% cents per core-hour, which is not much larger than the estimated 0.9 ± 25% cents per core-hour for the Fermilab data center.

From this work, we have shown that commercial cloud resources can be acquired at large scales for costs that are larger than, but comparable to, the cost of procuring and deploying similar resources on-site. Given the large year-over-year increases in the size of cloud computing industry-wide and the potential economies of scale, it is conceivable that the steady-state computing costs could approach or even undercut the price of procuring physical equipment. Beyond the comparison of steady-state costs, the needs and demands of the scientific community are not flat with respect to time, but have a structure and time-dependence.

References:

[1] Garzoglio G, these proceedings

[2] Gartner Group 2015 Magic Quadrant for Cloud Infrastructure as a Service, Worldwide

[3] Amazon Web Services 2016 AWS Offers Data Egress Discount to Researchers https://aws.amazon.com/blogs/publicsector/aws-offers-data-egress-discount-to-researchers/

[4] Girone M, Distributed Data Management and Distributed File Systems, *J. Phys.: Conf. Series*, **664.**

[5] Mhashilkar P, Tiradani A, Holzman B, Larson K, Sfiligoi I and Rynge M 2014 Cloud Bursting with GlideinWMS: Means to satisfy ever increasing computing needs for Scientific Workflows *J. Phys.: Conf. Series*, **513.**

[6] Wu H, Ren S, Timm S, Garzoglio G and Noh S-Y 2015 Experimental Study of Bidding Strategies for Scientific Workflows using AWS Spot Instances *8th IEEE Workshop on Many-Task Computing on Grids and Supercomputers*